

Pengenalan Greenfoot Dalam Pembuatan Game Berorientasi Objek

Syahbaniar Rofiah ^{1,*}

¹ Manajemen Informatika; STMIK Bina Insani; Jl.Raya Siliwangi No. 6 Rawa Panjang Sepanjang Jaya Bekasi, Telp. (021) 88958130. email: rsyahbaniar@gmail.com

* Korespondensi: e-mail: rsyahbaniar@gmail.com

Diterima: 18 Mei 2017; Review: 25 Mei 2017; Disetujui: 1 Juni 2017

Cara sitasi: Rofiah S. 2017. Pengenalan Greenfoot Dalam Pembuatan Game Berorientasi Objek. Information Management For Educators And Professionals. 1 (2): 213 – 222.

Abstrak: Perkembangan teknologi dan game saat ini semakin pesat sehingga muncul beraneka ragam jenis game. Game saat ini bukan hanya untuk hiburan semata akan tetapi menjadi kebutuhan sehari-hari. Sehingga bermunculan aplikasi-aplikasi sederhana dalam pembuatan game dan sangat memudahkan untuk pembuat game. Salah satunya aplikasi greenfoot yang mudah dioperasikan untuk pemula. Jika sebelumnya telah mempunyai dasar pemrograman java akan lebih mudah dalam pemahaman pemrograman didalam greenfoot. Bahasa yang dipergunakan dalam greenfoot adalah java. Java sendiri dikenal dengan pemrograman berorientasi objek. Cara mempelajari pemrograman berorientasi objek dalam bentuk animasi akan lebih mudah dimengerti dibandingkan dengan penjelasan teori.

Kata kunci: Greenfoot, Pemrograman Berorientasi Objek

Abstract: The development of technology and games is now growing rapidly so that the various kinds of games appear. The current game is not just for the mere entertainment but becomes a daily necessity. So popping up simple applications in game making and very easy for game makers.

One of them greenfoot applications that are easy to operate for beginners. If you have previously had a basic java programming will be easier in understanding programming in greenfoot. The language used in greenfoot is java. Java itself is known for object-oriented programming. How to learn object-oriented programming in the form of animation will be easier to understand compared with theoretical explanation.

Keywords: Greenfoot, Object Oriented Programming

1. Pendahuluan

Perkembangan teknologi dan game saat ini semakin pesat sehingga muncul beraneka ragam jenis game. Game saat ini bukan hanya untuk hiburan semata akan tetapi menjadi kebutuhan sehari-hari. Sehingga bermunculan aplikasi-aplikasi sederhana dalam pembuatan game dan sangat memudahkan untuk pembuat game.

Salah satunya aplikasi greenfoot yang mudah dioperasikan untuk pemula. Jika sebelumnya telah mempunyai dasar pemrograman java akan lebih mudah dalam pemahaman pemrograman didalam greenfoot. Bahasa yang dipergunakan dalam greenfoot adalah java. Java sendiri dikenal dengan pemrograman berorientasi objek. Cara mempelajari pemrograman berorientasi objek dalam bentuk animasi akan lebih mudah dimengerti dibandingkan dengan penjelasan teori.

Dengan menggunakan greenfoot pembuatan program akan lebih menyenangkan dan dapat dilakukan dengan mudah karena perintah-perintah yang digunakan tidak banyak. Sehingga memudahkan mahasiswa dan pengajar dalam memahami pemrograman tersebut.

1.1 Pengertian greenfoot

Greenfoot adalah perangkat lunak yang didesain untuk pemula agar dapat terbiasa dengan pemrograman berbasis objek (Object-Oriented Programming), yang mendukung pengembangan aplikasi bergambar dengan memakai pemrograman java (Irawan et al., 2015). Keuntungan dalam penggunaan greenfoot sebagai berikut:

- a. Merupakan software yang mudah untuk semua kalangan
- b. Software open source
- c. Menghemat proses waktu belajar mengajar
- d. Membuat siswa menyukai pemrograman

Kemudian kerugian dalam penggunaan greenfoot sebagai berikut:

- a. Game yang dihasilkan 2 dimensi dan tidak bisa 3 dimensi
- b. Perlu waktu mempelajari khususnya untuk siswa yang belum pernah belajar java

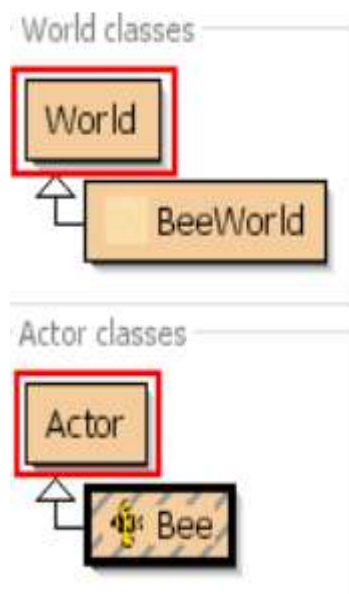
1.2 Class pada Greenfoot

Didalam greenfoot, sebuah kelas seperti bee (lebah) mendefinisikan karakteristik lebah yang berperan dalam scenario tersebut. Lebah alaminya mempunyai enam kaki dan 2 sayap kemudian lebah juga mewarisi ciri khas dalam hal warna dan bentuk ukuran tertentu.

1.3 Tipe Superclass

Didalam greenfoot tipe dari superclass ada 2 meliputi:

- a. World merupakan subclass yang mengatur gambar latar belakang dalam scenario dan mendefinisikan ukuran dan resolusi World tersebut.
- b. Actor merupakan yang bertindak dalam scenario tersebut.



Sumber: <http://www.academy.oracle.com>

Gambar 1. Tipe Superclass

1.4 Method

Method adalah seperangkat operasi atau tugas yang bisa dilakukan oleh class. Jika sebuah method dipanggil maka akan menjalankan operasi atau tugas melalui source code yang telah dibuat.

Method Name	Description
<code>void move(int distance)</code>	Assigns the object a number of steps to move, or the command to simply move when the Act or Run buttons are clicked.
<code>void turn(int amount)</code>	Assigns the object a number of degrees to turn.
<code>void act()</code>	Gives the object the opportunity to perform an action in the scenario. Method calls are inserted into this method.
<code>void setLocation(int x, int y)</code>	Assigns a new location for this object.
<code>void setRotation(int rotation)</code>	Sets a new rotation for this object.

Sumber: <http://www.academy.oracle.com>

Gambar 2. Method

2. Metode Penelitian

Tujuan dari penelitian ini adalah untuk memudahkan pengajar dan mahasiswa dalam memahami pemrograman berorientasi objek melalui greenfoot.

3. Hasil dan Pembahasan

3.1 Tahapan dalam Greenfoot

Pada saat membuka aplikasi greenfoot akan terlihat gambar 3 pada saat loading sebelum ke menu editor.

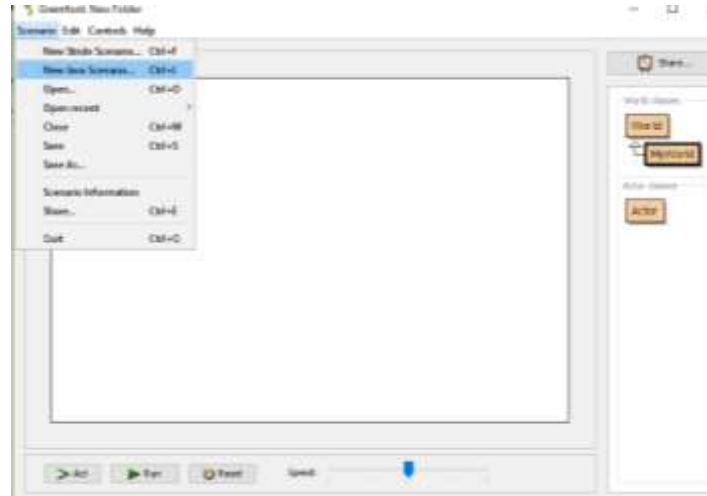


Sumber: <http://www.greenfoot.org>

Gambar 3. Halaman Muka Greenfoot

a. Membuat skenario baru

Untuk memulai projek baru langkah pertama dengan membuat skenario baru seperti terlihat pada gambar 4. Setelah membuat scenario baru akan terbentuk satu folder yang didalamnya terdapat dua folder images dan sounds serta file lain yang merupakan kesatuan dari skenario tersebut.



Sumber: <http://www.greenfoot.org>

Gambar 4. Halaman Muka Greenfoot

Terdapat kontrol eksekusi berfungsi untuk menjalankan skenario meliputi:

1. Act untuk menjalankan sekali dalam semua tindakan
2. Run/pause menjalankan semua tindakan dalam skenario berulang kali sampai dijeda
3. Reset menjeda skenario atau mereset skenario kembali keawal posisi
4. Speed menjalankan tindakan lebih cepat atau lebih lambat

b. Subclass (World)

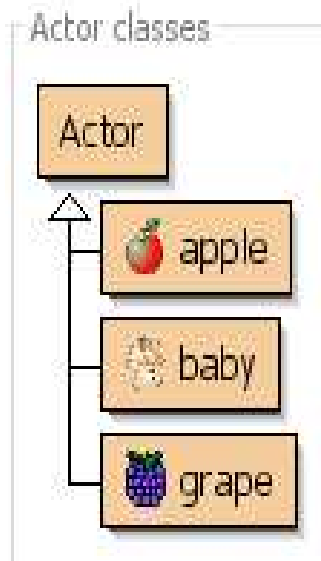
Mengubah background myworld sesuai yang dikehendaki dengan cara menyimpan gambar yang ingin digunakan kedalam folder images seperti terlihat pada gambar 5.



Sumber: <http://www.greenfoot.org>

Gambar 5. Halaman latar

Tahap selanjutnya setelah mengubah background kemudian menambahkan actor kedalam scenario. Dalam skenario pada gambar 5 terdapat 3 actor yaitu apple, baby dan grape yang masing-masing mempunyai tugas yang berbeda seperti terlihat pada gambar 6.

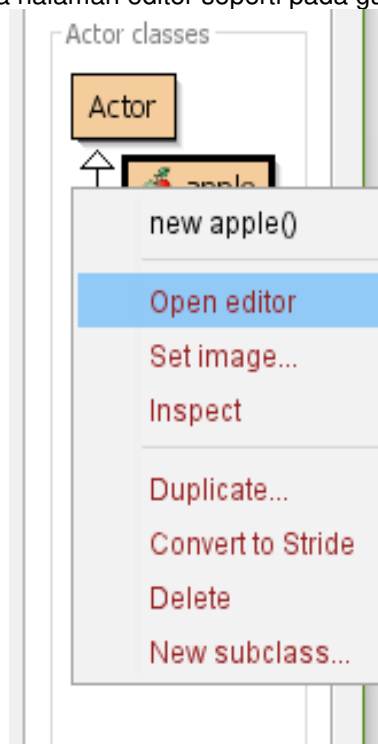


Sumber: <http://www.greenfoot.org>

Gambar 6. Actor

c. Menambahkan Perintah Dasar

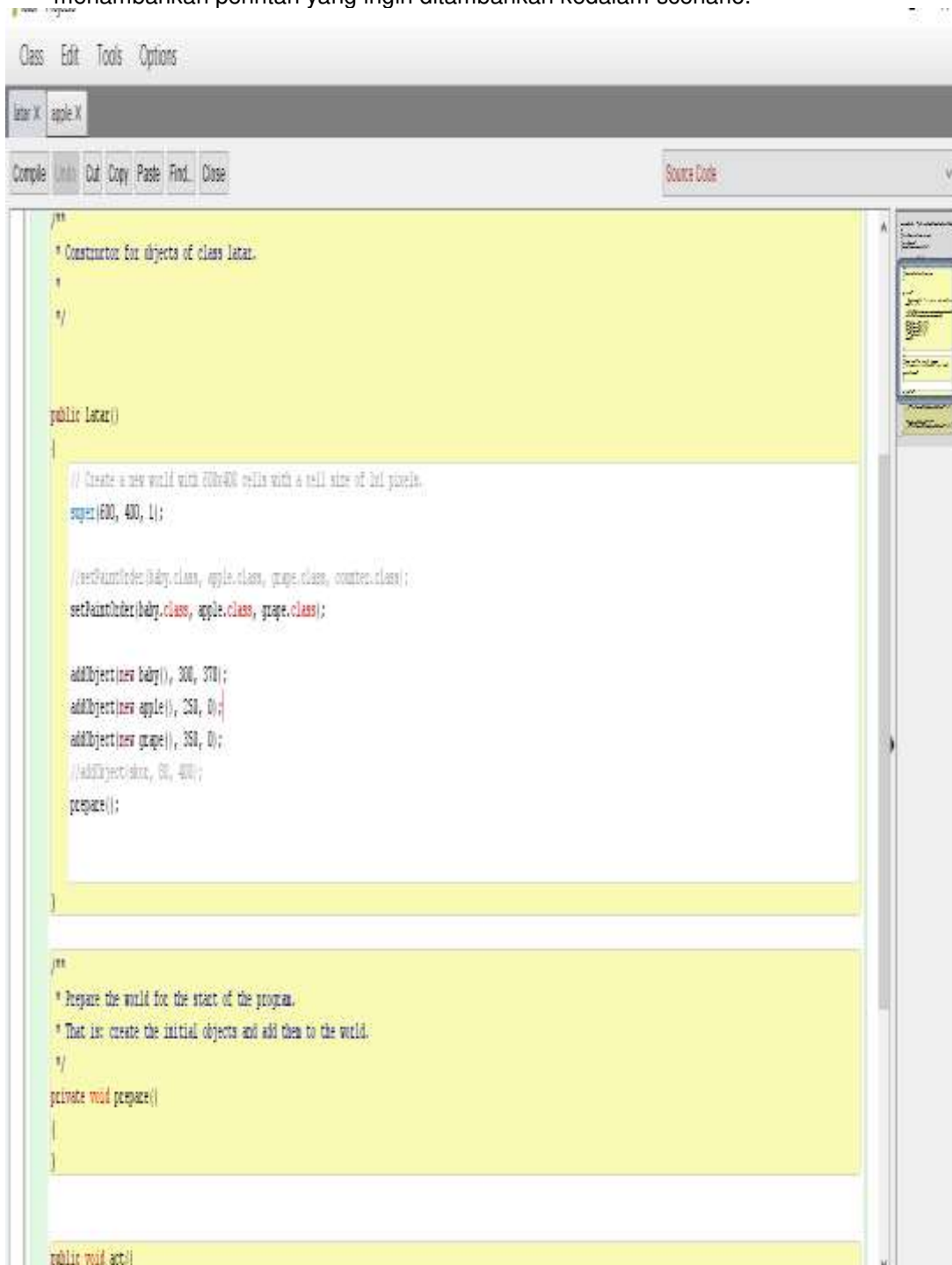
Dalam menambahkan perintah terlebih dahulu membuka halaman editor dengan cara klik kanan class untuk membuka halaman editor seperti pada gambar 7.



Sumber: <http://www.greenfoot.org>

Gambar 7. Membuka Editor

Kemudian akan tampil halaman editor seperti gambar 8. Langkah selanjutnya dengan menambahkan perintah yang ingin ditambahkan kedalam scenario.



Sumber: <http://www.greenfoot.org>

Gambar 8. Membuka Editor

d. Perintah getX()

Perintah getX() digunakan untuk menentukan titik koordinat dalam scenario pada actor seperti terlihat pada gambar 9.

```

public void act()
{
    //actor apple jatuh kebawah
    int x = getX();
    int y = getY();
    setLocation(x, y + 1);
    if (getY() == 500) //untuk apple dapat turun ke bawah
    {
        setLocation(getX(), 0); //jika apple sampai batas bawah, akan hilang
        // if (isTouching(apple.class)){
        //removeTouching(apple.class);
        getWorld().removeObject(this);
    }
}
}

```

Sumber: <http://www.greenfoot.org>

Gambar 9. Perintah getX()

e. Perintah getY()

Perintah getY() digunakan untuk menentukan titik koordinat dalam scenario pada actor seperti terlihat pada gambar 10.

```

public void act()
{
    //actor apple jatuh kebawah
    int x = getX();
    int y = getY();
    setLocation(x, y + 1);
    if (getY() == 500) //untuk apple dapat turun ke bawah
    {
        setLocation(getX(), 0); //jika apple sampai batas bawah, akan hilang
        // if (isTouching(apple.class)){
        //removeTouching(apple.class);
        getWorld().removeObject(this);
    }
}
}

```

Sumber: <http://www.greenfoot.org>

Gambar 10. Perintah getY()

f. Perintah isKeyDown()

Perintah ini digunakan untuk menjalankan perintah untuk keyboard pada saat ditekan seperti penggunaan tanda panah keatas, kebawah, kekanan dan kekiri seperti terlihat pada gambar 11.

```

if(Greenfoot.isKeyDown("Up")){
    setLocation(getX()-4,getY());
}

apple akan hilang
if(Greenfoot.isKeyDown("Right")){
    setLocation(getX()+1,getY());
}
else if (isTouching(apple.class)){
    removeTouching(apple.class);
}

if(Greenfoot.isKeyDown("Left")){
    setLocation(getX()-1,getY());
}

```

Sumber: <http://www.greenfoot.org>

Gambar 11. Perintah isKeyDown()

g. Perintah setLocation()

Perintah setLocation digunakan untuk menentukan titik koordinat antara sumbu x dan sumbu y seperti terlihat pada gambar 12.

```

public void act()
{
    //actor grape jatuh kebawah
    int x = getX();
    int y = getY();
    setLocation(x, y + 1);

    if (getY() == 499)
    {
        setLocation(getX(),0);
    }
}

```

Sumber: <http://www.greenfoot.org>

Gambar 12. Perintah setLocation()

h. Perintah addObject()

Perintah **addObject()** digunakan untuk menambahkan objek baru pada world. Perintah **addObject** ini biasanya digunakan pada saat runtime seperti terlihat pada gambar 13.

```
public latar()
{
    // Create a new world with 600x400 cells with a cell :
    super(600, 400, 1);

    //setPaintOrder(baby.class, apple.class, grape.class,
    setPaintOrder(baby.class, apple.class, grape.class);

    addObject(new baby(), 300, 370);
    addObject(new apple(), 250, 0);
    addObject(new grape(), 350, 0);
    //addObject(skor, 80, 400);
    prepare();
}
```

Sumber: <http://www.greenfoot.org>

Gambar 13. Perintah addObject()

i. Perintah removeTouching()

Perintah **removeTouching()** digunakan untuk menghilangkan objek dari world. Perintah **removeTouching** ini biasanya digunakan pada saat runtime seperti terlihat pada gambar 14.

```
if(Greenfoot.isKeyDown("Right")){
    setLocation(getX()+1,getY());
}
else if (isTouching(apple.class)){
    removeTouching(apple.class);
}

if(Greenfoot.isKeyDown("Left")){
    setLocation(getX()-1,getY());
}
else if (isTouching(apple.class)){
    removeTouching(apple.class);
}

//grape akan hilang
if(Greenfoot.isKeyDown("Right")){
    setLocation(getX()+1,getY());
}
else if (isTouching(grape.class)){
    removeTouching(grape.class);
}
```

Sumber: <http://www.greenfoot.org>

Gambar 14. Perintah removeTouching()

4.Kesimpulan

Kesimpulan yang dapat dicapai bahwa cara pemrograman berorientasi objek menggunakan greenfoot dalam bentuk animasi akan lebih dimengerti dibandingkan dengan

penjelasan teori. Kurangnya pemahaman tentang bahasa yang digunakan pada greenfoot akan mempersulit kelanjutan dalam pembuatan kode perintah-perintah selanjutnya.

Referensi

Irawan JD, Prasetio S, Wibowo SA, Pranoto YA. 2015. Pelatihan Pembuatan Game Menggunakan Greenfoot. Jurnal Industri Inovatif Vol 5 No 2 September 29-35.
<http://ejournal.itn.ac.id/index.php/industri/article/download/85/79/>

Kölling M, 2009, *Introduction to Programming with Greenfoot*, Pearson Education

Kölling M, Poul Henriksen, *greenfoot: Combining Object Visualisation with Interaction*

<http://www.academy.oracle.com>

<http://www.greenfoot.>